

## **REMARKS**

Applicants express appreciation to the Examiner for the recent interview held with applicants' representative. As presented herein for reconsideration, the claims have been amended as proposed at the interview. Specifically, claims 1, 3, 4, 8, 11 – 15, 17 – 21 and 23 have been amended<sup>1</sup>, claims 2, 5 – 7 and 24 – 30 have been cancelled without prejudice, and new claim 31 has been presented. Thus, by this paper, claims 1, 3, 4, 8 – 23 and 31 are pending and presented for reconsideration, of which claims 1 (directed to a method) and 31 (directed to a corresponding computer program product) are the independent claims.

As noted by applicants in their specification, when a user is using a design application such as BizTalk Orchestration Designer™, displaying a limited number of messages when encountering configuration parameter errors so that the errors are grouped under common errors allows the designer to more quickly identify and correct the configuration errors. Thus, the "design experience is improved when a user is presented with an abbreviated list of the most important proposed solutions rather than an exhaustive list of all proposed solutions. When presented with multiple proposed solutions, the user must first determine if there is an association between the proposed solutions or a common cause." P. 8 ¶ 27. Thus, automatically identifying a common error for an entire group of errors arising from inconsistent configuration parameters of different software components in accordance with applicants' claimed method and computer program product for implementing the method, significantly simplifies the design process.

As presented herein for reconsideration, the independent claims are directed to a method of alerting a user/software designer to configuration errors that may arise due to inconsistencies between the configuration parameters of one or more of the selected software components for the software design, and a computer program product for implementing the method. The claimed method is implemented in a computing system which includes a visual design surface in the form of a user interface having a plurality of shapes that are selectable by a user, each shape being associated with one or more configuration parameters that define characteristics of software components such as relative positioning of shapes, connections between shapes and other design

---

<sup>1</sup> Any amendments to claims other than those which are expressly relied upon in overcoming the rejections on art have been made simply to insure consistency in claim language, to correct typographical or grammatical errors, or to correct other errors of a formal, non-substantive nature, but not to otherwise narrow the claims in scope for any reason.

parameters set by a user, and wherein a user may select and arrange a plurality of the shapes when designing a software application by dragging and dropping them to a design section of the interface. As claimed, the method is comprised of the user selecting a first shape and copying it to the design section of the interface, and then selecting a second shape and copying it to the design section of the interface so that the second shape is functionally interactive with the first shape. Next, the computing system automatically evaluates with a configuration module the configuration parameters of the first and second shapes by accessing a configuration rules database which is used by the configuration evaluation module to determine whether the configuration parameters of the two functionally interactive shapes in the proposed design violate any configuration rules. When a configuration rule is determined to have been violated, the computing system then automatically evaluates any errors using an error module that accesses a common error database which defines at least some errors as being grouped under a common error, and then returns a common error for display to the designer in order to limit the number of error messages presented to the user during the design process. An icon is displayed by the computing system next to at least one of the first or second selected shapes to represent at least one common configuration error associated with the shape, so that in response the designer selects the icon, displaying at least one proposed solution to the common configuration error presented.

In the Office Action the claims were rejected solely on grounds that they were found to be obvious under 35 U.S.C. § 103(a) over U.S. Pat. No. 6,225,998 (Okita et al.) as further modified by teachings found in U.S. Pat. No. 7,165,194 (Paradkar).<sup>2</sup>

Applicants claimed method and computer program product as presented herein for reconsideration are significantly different from the prior art or record and are neither anticipated nor made obvious by the prior art. Okita et al. is directed to a visual design application that relates to improvements in customer transaction processing systems, particularly the visual design of transaction flows. Col. 1 lines 5 – 10. In existing systems, as noted by Okita et al., “generation of routing tables or routing procedures to control transaction flows requires computer programming skills.” Recognizing this problem, Okita et al. describe a method for

---

<sup>2</sup> Since Paradkar qualifies as “prior” art, if at all, under 35 U.S.C. 102(e) applicants reserve the right to challenge the status of that reference as qualifying “prior” art. Accordingly, any statement or comment herein to Paradkar is made merely for purposes of argument, and assumes *arguendo* that the reference is proper qualifying prior art.

displaying visual primitives of transaction flow used by a transaction processing system as a means by which development of routing tables or routing procedures for customer transaction processing is simplified and better coordinated across distributed environments. Col. 1 lines 49 – 66 and col. 2 lines 1 – 7. When using the method of Okita et al. to prepare a visual representation of a customer transaction process, “If a particular workflow is not complete, or contains errors, a visual indicator (or visual alert) is generated and displayed by the workflow editor.” Col. 15 lines 48 – 50.

As noted in applicants’ response of Dec. 11, 2006, and as acknowledged in the present Office Action,<sup>3</sup> “the cited sections of Okita . . . make no mention of ‘displaying at least one proposed solution to a configuration error,’ as claimed.”

Paradkar, on the other hand, is directed to “A computer software program used to capture configuration values of predetermined parameters of vendor software and an operating system on which the vendor software is installed. . . . A collateral report of configuration values for the customer computer systems is sent to the vendor for diagnosis of the technical problem experienced by the customer.” Abstract. “The support program can be programmed to report configuration errors in the customer environment, and suggest possible solutions.” Col. 9 lines 22 – 24.

Apart from whether Paradkar is properly combinable with Okita et al. as asserted,<sup>4</sup> and perhaps more importantly, neither Okita et al. nor Paradkar are directed to simplifying and improving the design experience of a software designer by

“automatically evaluating with a configuration module the configuration parameters of the first and second shapes by accessing a configuration rules database which is used by the configuration evaluation module to determine whether the configuration parameters of the two functionally interactive shapes in the proposed design violate any configuration rules;

when a configuration rule is determined to have been violated, automatically evaluating any errors using an error module that accesses a common error database which

---

<sup>3</sup> Office Action p. 3 (“Okita fails to explicitly disclose a method of, displaying at least one proposed solution to a configuration error.”).

<sup>4</sup> Applicants reserve the right to further challenge whether the Examiner’s proposed combination of prior art is properly combinable if and as such becomes a necessary issue to resolution of the rejections, which is not the case since in any event the references fail to teach or suggest the claimed method, as noted above.

defines at least some errors as being grouped under a common error, and then returning a common error for display to the user in order to limit the number of error messages presented to the user during the design process;

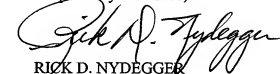
displaying an icon next to a at least one of the first or second selected shapes to represent at least one common configuration error associated with the shape; and in response to a user selecting the icon, displaying at least one proposed solution to the common configuration error presented." Claims 1 and 31.

As noted in the Interview Summary at the conclusion of the Interview, "The proposed amendments appear to overcome the prior art rejection of record." Thus, for at least the reasons noted, the claims as presented herein for reconsideration are patentable over the prior art and thus favorable action is courteously requested.

In the event the Examiner finds any remaining impediment to allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 25<sup>th</sup> day of July, 2008.

Respectfully submitted,

  
RICK D. NYDEGGER  
Registration No. 28,651  
Attorney for Applicant  
Customer No. 047973